2425-CT2106 – Assignment 1 ID:23344296

Q.1

- Class 1: Light
  Boolean isOn – binary
  String colour ← return colour
  setOn setOff states
  PrintState()

- Class 2: TrafficLights
  Assigning three different colours, object fields: Green Amber Red
  Behaviour methods: go, prepareToStop, Stop, printState

- Class 3: TestTrafficLights

For loop containing TrafficLights methods

Relationships between classes: The light class is an object that is required in the Traffic Lights class 3 times per traffic cycle. Each state of the lights (go, prepare to stop, stop) requires a trio of Light object with their respective Boolean states. Test Traffic Lights then takes these states and prints them in the correct order. This composition of classes has a linear dependency.

Q.2

LIGHT CLASS

```
public class Light
{
   //initialising objects
   private String color;
   private boolean isOn;

   //color being passed to constructor of class
```

```
Light(String color){
   this.color = color;
   this.isOn = false;
}


//accessor and mutator method for the color field
 //<Visibility> <ReturnType> <MethodName> ( <Params?> )

public String getColor(){

   return color;
}


public void setColor(String cL){

   color = cL;
}

//creating light on function
public void setOn(){

   isOn = true;
}

//creating light off function
public void setOff(){
```

```
      isOn = false;
   }


   //Logic of either Light on or Light off print output
   public void printState(){
     if(isOn==true){
        System.out.println(color);
     }
     else{
        System.out.println("=OFF=");
     }

   }
}
```

TRAFFICLIGHTS CLASS

```
public class TrafficLights{

   Light green = new Light("[ Green ]");
   Light amber = new Light("[ Amber ]");
   Light red = new Light("[ Red ]");

   //"go" function and corresponding light pattern
   public void go(){

      green.setOn();
```

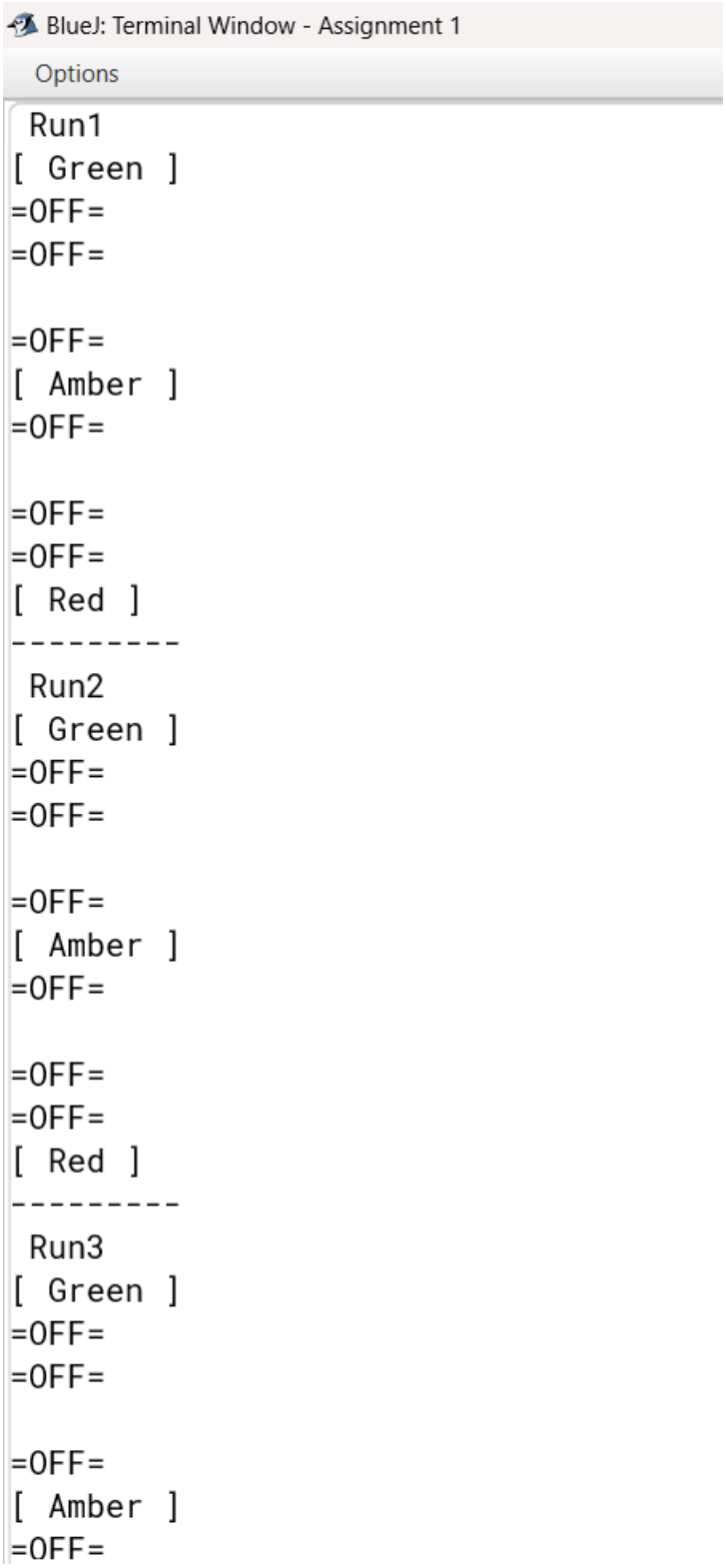
```
    amber.setOff();
    red.setOff();
}

//"prepare to stop" function and corresponding light pattern
public void prepareToStop(){

    green.setOff();
    amber.setOn();
    red.setOff();
}

//"stop" function and corresponding light pattern
public void stop(){

    green.setOff();
    amber.setOff();
    red.setOn();
}


// printing given on/off status of each light
public void printState(){

    green.printState();
    amber.printState();
    red.printState();
```

```
    }
}
```

TESTTRAFFICLIGHT CLASS

```
public class TestTrafficLight
{
    public static void main(String[]args){

        TrafficLights lights = new TrafficLights();

    // for loop for 5 runs
    int n = 5;
     for(int i = 1; i<= n; ++i){

        System.out.println(" Run"+i);
        lights.go();
        lights.printState();
        System.out.println("");
        lights.prepareToStop();
        lights.printState();
        System.out.println("");
        lights.stop();
        lights.printState();
        System.out.println("---------");

     }
    }
}
```

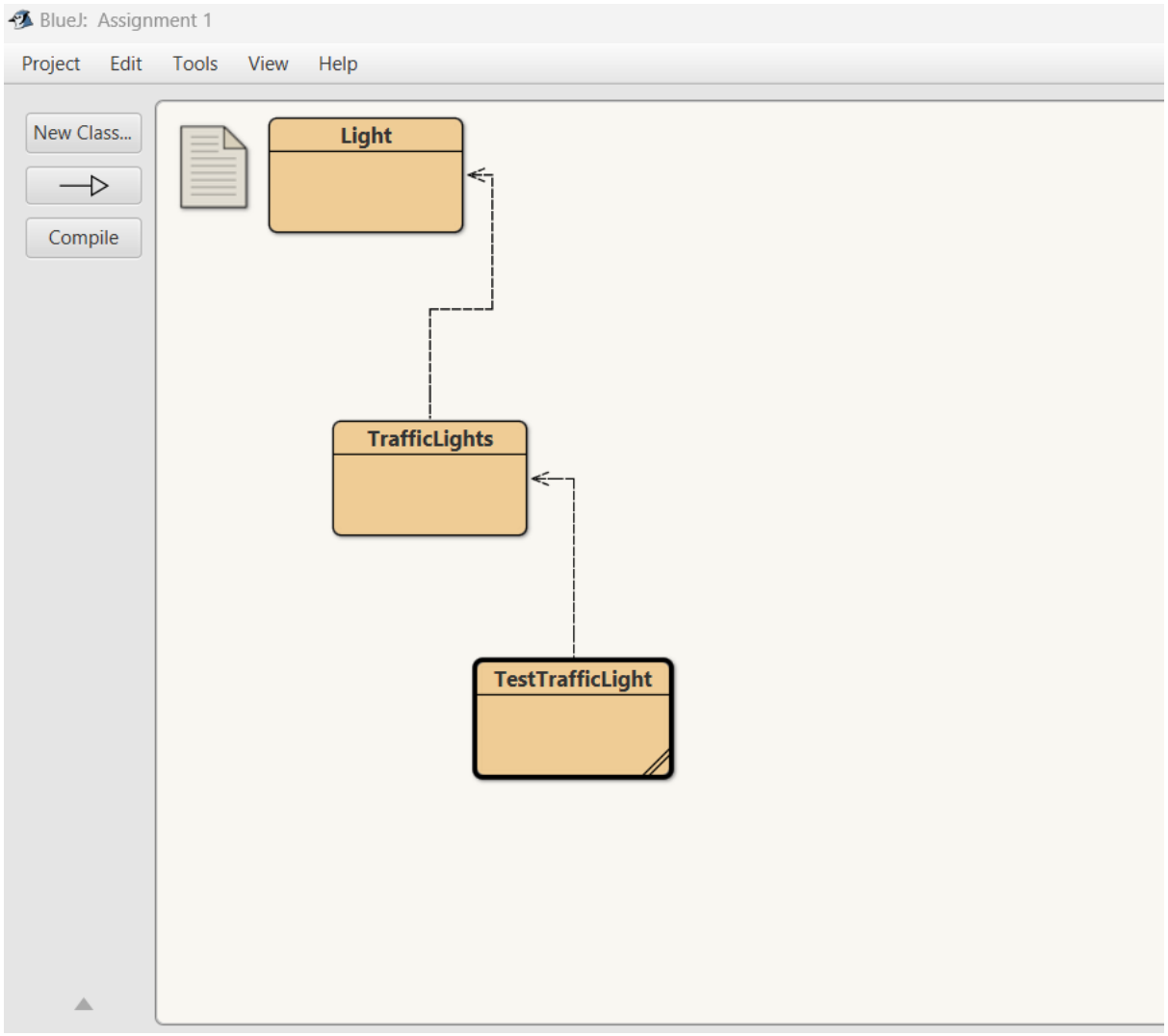
Q.3


BlueJ: Terminal Window - Assignment 1
Options
Run1
[Green]
=OFF=
=OFF=
=OFF=
[Amber]
=OFF=
=OFF=
=OFF=
[Red]

Run2
[Green]
=OFF=
=OFF=
=OFF=
[Amber]
=OFF=
=OFF=
=OFF=
[Red]

Run3
[Green]
=OFF=
=OFF=
=OFF=
[Amber]
=OFF=

```
[ Amber ]
=OFF=

=OFF=
=OFF=
[ Red ]
---------
 Run4
[ Green ]
=OFF=
=OFF=

=OFF=
[ Amber ]
=OFF=

=OFF=
=OFF=
[ Red ]
---------
 Run5
[ Green ]
=OFF=
=OFF=

=OFF=
[ Amber ]
=OFF=

=OFF=
=OFF=
[ Red ]
---------
```


BlueJ: Assignment 1
Project
Edit
Tools
View
Help
New Class...
Compile
Light
TrafficLights
TestTrafficLight